

Introducing
The OSI Gazette

The 6502 Resource Magazine
PET•Apple•Atari•OSI•KIM•SYM•AIM

Hard Disks
For The Apple

COMPUTE!

The Journal for Progressive Computing™

\$2.00
November/
December,
1980
Issue 7
Vol. 2, No.6

COMPUTE!
Looks At
The New TRS-80
Color Computer?

Times Square On
Your Atari

Interfacing
KIM/SYM/AIM/
OSI With BASIC

Combining
BASIC And
Machine
Language, II

Visible Memory
PET Printer Dump

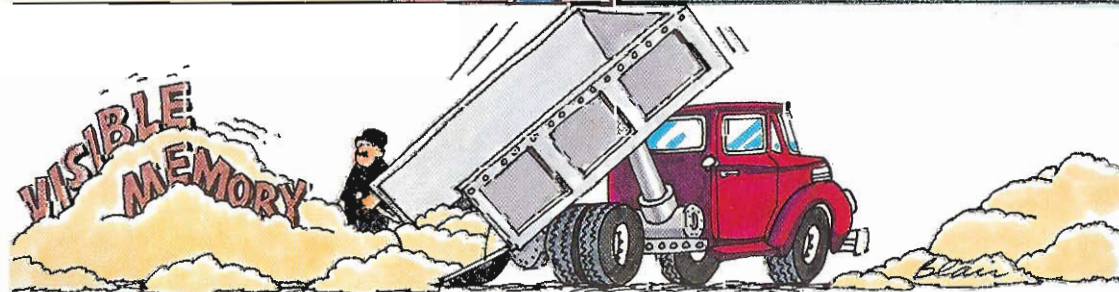


Table of Contents

November/December 1980, Issue 7, Vol. 1, No. 6

The Editor's Notes Robert C. Lock, 4
 The Reader's Feedback Robert Lock and Readers, 8
 Computers & Society David D. Thornburg and Betty J. Burr, 10
 Music And The Personal Computer Frank Winter, 18
 Photos by John Wood, Toronto

Micros With The Handicapped Susan Semancik, 22
 Small Computers And Small Libraries Arthur L. McNeil, 24
 Efficiency With Subroutines Mike Richter, 30
 Computing Correlation Coefficients Brian J. Flynn, 36

The Apple Gazette 42
 Al Baker's Programming Hints: APPLE Al Baker, 42
 The Anatomy of A Word Research Processing Program
 For The Apple Derek A. Kelley, 44
 Hard Disks For The Apple Philip Castevens, 50

The Atari Gazette 56
 Times Square On Your Atari Neil Harris, 56
 Error Reporting System For The Atari Len Lindsay, 58
 An Atari BASIC Tutorial: Monthly Bar Graph Program Jerry White, 61
 Card Games In Graphics Modes 1 and 2 William D. Seivert, 62
 Using TAB In Atari BASIC James L. Bruun, 64
 Pokin' Around Charles Brannon, 66
 Winning Star Raiders William L. Colsher, 68
 Logo Generation Wynn Smith, 69

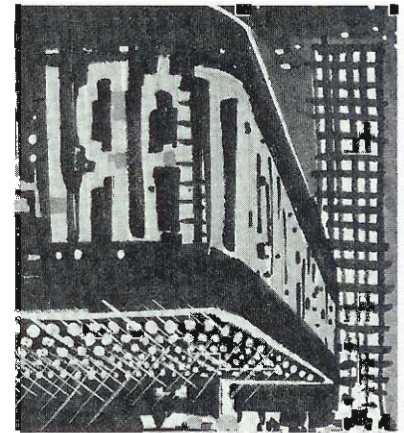
The OSI Gazette 70
 Coded Data For OSI CIP Charles Stewart, 70
 OSI Graphics Character Set W. Blaine Garland, 71
 Atari Joysticks On The OSI CIP Charles L. Stanford, 72
 Correction Listing For Fast Graphics On The OSI CIP Charles L. Stanford, 76
 Breakout For OSI CIP Charles Stewart, 77

The PET Gazette 78
 Basic CBM 8010 Modem Routines Jim Butterfield, 78
 Programmer's Notes For The CBM 8032 Roy Busdiecker, 80
 Keyprint Charles Brannon, 84
 PET 4.0 ROM Routines Jim Butterfield, 88
 BASIC 4.0 Memory Map Jim Butterfield, 92
 Algebraic Expression Input For The PET, Version 2 Elizabeth Deal, 94
 Defining A Function Whilst Running A Program M. J. Winter, 96
 Machine Language: Addressing Modes Jim Butterfield, 98
 Visible Memory Printer Dump Dr. Frank Covitz, 104
 Disk Lister: A Disk Cataloging Program Robert Baker, 110
 Commodore Dealers Form Cooperative Joretta Klepfer, 115

The SBC Gazette 116
 Nuts and Volts Gene Zumchak, 116
 Interfacing The Am9511 Arithmetic Processing Unit Marvin L. DeJong, 122
 Interfacing KIM/SYM/AIM/OSI With BASIC Jim Butterfield, 128
 Review: 6502 Games; SYBEX Games Board (for SYM) Harvey B. Herman, 132
 KIM-1 Tidbits Harvey B. Herman, 134
 AIM 65 Tape Copy Utility Christopher J. Flynn, 137
 Combining BASIC and Machine Language Programs On Tape George Wells, 140
 SYM (AIM) Hi-Speed Tape Revisited Gene Zumchak, 142

CAPUTE! Wherein We Acknowledge Recent Goofs 144

Short Programs:
 Base Converter: (Decimal to Hex) Fred D. Bean, 144
 An Atari Random Character Generator Paul Dobosz, 148
 A First Look At The TRS-80 Color Computer David D. Thornburg, 150
 Index to Advertisers 152
 Pet Program Listing Conventions 152



Page 56



Page 70



Page 104



Page 128

COMPUTE. The Journal for Progressive Computing (USPS: 537250) is published six times each year by Small System Services, Inc., P.O. Box 5406, Greensboro, NC 27403 USA. Phone: (919) 275-9809. Editorial Offices are located at 200 East Bessemer Ave., Greensboro, NC 27401.

Domestic Subscriptions: 12 issues, \$16.00. Send subscription orders or change of address (P.O. Form 3579) to Circulation Dept., COMPUTE. Magazine, P.O. Box 5406, Greensboro, NC 27403. Controlled circulation postage paid at Greensboro, NC 27403. Entire contents copyright © 1980 by Small System Services, Inc. All rights reserved. ISSN 0194-357X.

Disk Lister

A Disk Cataloging Program for the Commodore Pet and 2040 Disk

Baker Enterprises
15 Windsor Drive,
Atco, New Jersey 08004

Having finally copied all my programs from cassette onto floppy disks, I suddenly found it somewhat difficult to find out where anything was. With well over 300 programs scattered onto 20 or 30 disks, it just wasn't easy to quickly locate a particular program. In addition, I was starting to use Word Pro 3 quite heavily to write articles and various documents, saving them all on disk as well. Because of this, I decided to write a program to catalog all the disks and condense the information onto a single diskette.

The program shown here is the first step toward my final goal. It can catalog well over 100 diskettes with the current Commodore 2040 disk drive. It only has a few functions implemented, but it has proven to be very handy. I have a "wish" list of other features I intend to add in the near future. All I need now is the time to do it!

The major flow of the program should be straight forward. I've sprinkled the program with REMarks to help document several operations and a few of the variables used. If you should copy the program, I would strongly recommend leaving out all REMarks and unnecessary spaces to help speed up program execution.

In its present form, the program reads the directory of any disk placed in drive #1. It then writes a condensed directory as a data file on the master directory disk in drive #0. All of this is done automatically without any user input other than selecting the program function and verifying the correct disk was inserted. Once the data files are created, you can then display or print the directory of any disk that has been cataloged in the master directory. The directory will show the disk name, ID, and format. It will also show an alphabetized list of the files on the disk along with the file type and length (in blocks) of each file. While a directory is being listed, hitting "S" will stop the listing until another key is hit. Hitting "Q" at any time during the listing will terminate the list function. A sample directory printout is shown in Figure 1 to give you an idea of what is displayed.

The file names of the sequential data files created for the master directory consist of the two character disk ID followed by a period and the letters DIR. In its compacted form, the major disk information takes 25 bytes and each entry in the directory takes 20 bytes. Since the disk ID is used to create the

data file name, be careful not to duplicate disk ID's. This precaution is also recommended when upgrading to DOS 2.0 since DOS uses the ID to recognize a disk has been changed in the drive. Another hint on using this program - reserve one disk as the master directory disk with nothing else stored on that disk except the directory data files. This will allow cataloging the maximum number of disks into your master directory.

If a cataloged disk is later updated or modified, simply re-catalog the disk to update the master directory. The old data file will be deleted and a new one created, all automatically. The program also provides a delete function, so you can delete a cataloged disk that no longer exists. This function simply deletes the appropriate data file for the specified disk ID. You could actually accomplish the same function by manually scratching the correct data file from the master directory disk.

Currently, when listing or deleting directories, you must enter the two character disk ID. This can be inconvenient at times, but it does make things easier. I intend to allow entering the ID -or- the disk name in the next version I'm working on. However, this will require maintaining some kind of cross-reference to correlate the disk ID's and disk names. When this feature is added, the delete function will always have to be used to remove a disk from the master directory. The added cross-reference will also be the basis for several other new features I intend to add:

List all disk ID's currently used in alphabetical order; optionally display each disk's corresponding 16 character name. This will help avoid using duplicate disk IDs when creating new disks.

List all disk names in alphabetical order and show each disk's corresponding 2-character disk ID.

Ability to list all disks a particular file can be found on. This function should use character matching in case you can't remember the exact file name or want all files starting with a particular word, etc.

One other thing I would like to add is computation of the number of free blocks from the BAM. If this information were included in the data files for each disk, you could then list all disks with the number of free blocks displayed. This would allow quickly finding space on a disk to save a new program of known length.

Right now I'm not sure when I'll be able to get around to finishing this project. At least I've got something useful for now and it does help tremendously. If you have any ideas or suggestions as to other features you think might be useful, or if you're interested in how the final version turns out, let me know.


```

10 REM ***** DISK LISTER *****
20 REM
30 REM      BY: ROBERT W. BAKER
40 REM
50 REM 15 WINDSOR DRIVE, ATCO, NJ 08004
60 REM
70 REM *****
80 :
90 CLR: DIM D$(150), D(150): Q$=CHR$(34):
  -CR$=CHR$(13)
100 REM DISPLAY MENU & SELECT FUNCTION
110 PRINT"â";SPC(9);"ID I S K  L I S T -
  -E R":GOSUB 1340
120 PRINT SPC(5);"0 - DONEâ
130 PRINT SPC(5);"1 - UPDATE MASTER -
  -DIRECTORYâ
140 PRINT SPC(5);"2 - DISPLAY SELECTED -
  -DIRECTORYâ
150 PRINT SPC(5);"3 - DELETE DISK ENTRY -
  -FROM MASTER
160 GOSUB 1340
170 PRINT"âENTER DESIRED FUNCTION: ";
180 GOSUB 1320
190 IF C$="0" THEN PRINT"â": END
200 C=VAL(C$):IF C<1 OR C>3 THEN 180
210 ON C GOTO 250,750,1050
220 REM *****
230 REM UPDATE MASTER DIRECTORY
240 REM *****
250 PRINT"âINSERT UPDATE DISK IN DRIVE -
  -#1
260 GOSUB 1310:GOSUB 1340:PRINT"OK
270 OPEN 15,8,15
280 PRINT#15,"I1"
290 OPEN 5,8,5,"$1,S,R":GOSUB 1260
300 Y=142:GOSUB 1200:REM *** SKIP BAM
310 Y=16:GOSUB 1180:DN$=S$:REM *** DISK -
  -NAME
320 Y=2:GOSUB 1200:REM *** SKIP SPACES
330 Y=2:GOSUB 1180:DI$=S$:REM *** DISK -
  -ID
340 PRINT"âIDISK NAME:â ";DN$:
  -PRINT"âIDISK ID:â "DI$:
  -GOSUB 1340
350 PRINT"CORRECT DISK INSERTED";:
  -GOSUB 1350:IF C$="N" THEN 710
360 GOSUB 1340:PRINT"READING DIRECTORY -
  -ENTRIES...
370 GOSUB 1250
380 Y=2:GOSUB 1180:DF$=S$:REM *** DISK -
  -FORMAT
390 Y=89:GOSUB 1200:NF=0:Z=0:REM *** -
  -SKIP TO FIRST DIRECTORY ENTRY
400 GOSUB 1220:FT=V:F$=C$:REM *** FILE -
  -TYPE (0=DELETED)
410 Y=2:GOSUB 1200:REM *** SKIP -
  -STARTING TRACK & SECTOR
420 Y=16:GOSUB 1180:REM *** FILE NAME
430 Y=9:GOSUB 1200:REM *** SKIP UNUSED -
  -INFO
440 GOSUB 1220:X=V:GOSUB 1220:X=X+(V*256
  -):REM *** #BLOCKS IN FILE
450 IF FT>0 THEN NF=NF+1:D$(NF)=F$+S$:
  -D(NF)=X:REM *** ADD FILE IF NOT -
  -DELETED
460 Z=Z+1:Z=Z-(INT(Z/8)*8):REM *** -
  -Z=ENTRY WITHIN THIS DISK BLOCK
470 IF Z>0 THEN Y=2:GOSUB 1200:REM ** -
  -SKIP 2 BYTES IF NOT LAST ENTRY IN -
  -BLOCK
480 IF SS=0 THEN 400:REM *** CONTINUE -
  -TILL END OF DIRECTORY
490 CLOSE 5:IF NF<2 THEN 600
500 GOSUB 1340
510 PRINT"SORTING DIRECTORY ENTRIES...
520 REM SORT DIRECTORY INTO
530 REM ALPHABETICAL ORDER
540 FOR X=1 TO NF:FOR Y=1 TO NF-1
550 IF D$(Y)<=D$(Y+1) THEN 570
560 C$=D$(Y):C=D(Y):D$(Y)=D$(Y+1):
  -D(Y)=D(Y+1):D$(Y+1)=C$:D(Y+1)=C
570 NEXT Y,X
580 REM DELETE OLD DIRECTORY
590 REM DATA FILE & SAVE NEW COPY
600 GOSUB 1340:PRINT"UPDATING MASTER -
  -DIRECTORY...
610 S$="0:"+DI$+".DIR"
620 PRINT#15,"S"+S$
630 OPEN 5,8,5,S$+",S,W":GOSUB 1260
640 PRINT#5,Q$:DN$:Q$:CR$:GOSUB 1260
650 PRINT#5,DI$:CR$:GOSUB 1260
660 PRINT#5,DF$:CR$:GOSUB 1260
670 IF NF=0 THEN 710
680 FOR X=1 TO NF:FOR Y=1 TO 17:
  -PRINT#5,MID$(D$(X),Y,1):GOSUB -
  -1260:NEXT Y
690 H=INT(D(X)/256):L=D(X)-(256*H)
700 PRINT#5,CHR$(L);CHR$(H);CR$:;
  -GOSUB 1260:NEXT X
710 CLOSE 5:CLOSE 15:GOTO 110
720 REM *****
730 REM DISPLAY SELECTED DISK DIRECTORY
740 REM *****
750 PRINT"âTO DISPLAY DISK DIRECTORY":
  -GOSUB 1140:OPEN 15,8,15
760 OPEN 5,8,5,S$+",S,R":GOSUB 1260
770 GOSUB 1340:PRINT"WANT PRINTED -
  -COPY":GOSUB 1350:GOSUB 1340
780 PD=3:IF C$="Y" THEN PD=4
790 OPEN 4,PD:REM *** PD = PRINT DEVICE -
  -SELECTOR (3=DISPLAY, 4=PRINTER)
800 INPUT#5,DN$:GOSUB 1260
810 INPUT#5,DI$:GOSUB 1260
820 INPUT#5,DF$:GOSUB 1260
830 IF PD=3 THEN PRINT"â";
840 PRINT#4,"âIDISK NAME:â ";DN$
850 PRINT#4
860 PRINT#4,"âIDISK ID:â ";DI$:SPC(10);"
  -âIDISK FORMAT:â ";DF$
870 PRINT#4:REM *** DISK FORMAT WILL -
  -BE BLANK FOR DOS 1.0
880 PRINT#4,"CCCCCCCCCCCCCCCCCCCCCCCCCCCC
  -CCCCCCCCCCCC":PRINT#4
890 Y=17:GOSUB 1180:REM *** GET FILE -
  -NAME & TYPE
900 GOSUB 1220:Z=V:GOSUB 1220:Z=Z+(256*V
  -):REM *** GET #BLOCKS
910 GOSUB 1250:REM *** SKIP LAST CR
920 PRINT#4,RIGHT$(" "+STR$(Z),
  -4);" ";
930 PRINT#4,MID$(S$,2,16);SPC(3);
940 V=ASC(LEFT$(S$,1)):REM *** DECODE -
  -FILE TYPE
950 IF V=129 THEN PRINT#4,"SEQ";

```

```

960 IF V=130 THEN PRINT#4,"PGM";
970 IF V=131 THEN PRINT#4,"USR";
980 PRINT#4:GET C$:IF C$="S" THEN GOSUB ↵
    ↵1320:REM *** ALLOW START/STOP OF ↵
    ↵LIST
990 IF C$<>"Q" AND SS=0 THEN 890
1000 CLOSE 4:CLOSE 5:CLOSE 15:IF PD=3 ↵
    ↵THEN GOTO 1300
1010 GOTO 110
1020 REM *****
1030 REM DELETE DISK DIRECTORY DATA FILE
1040 REM *****
1050 PRINT"↵TO DELETE DISK FROM MASTER ↵
    ↵-DIRECTORY":GOSUB 1140:OPEN 15,8,15
1060 PRINT#15,"S"+S$:CLOSE 15:GOTO 110
1070 :
1080 REM *****
1090 REM ***** SUBROUTINES *****
1100 REM *****
1110 :
1120 REM *** GET DISK ID
1130 REM *** & MAKE DATA FILE NAME
1140 INPUT"↵ENTER DISK ID  ↵<<<";DI$
1150 IF DI$="↵" THEN 110
1160 S$="0:"+LEFT$(DI$,2)+".DIR":RETURN
1170 REM *** READ STRING FROM DISK,
    ↵ Y-BYTES LONG
1180 S$="":FOR X=1 TO Y:GOSUB 1250:
    ↵S$=S$+C$:NEXT X:RETURN
1190 REM *** SKIP Y-BYTES OF DISK FILE
1200 FOR X=1 TO Y:GOSUB 1250:NEXT X:
    ↵RETURN
1210 REM *** READ BYTE & RETURN ASC ↵
    ↵-VALUE
1220 V=0:GOSUB 1250:IF C$<>"↵" THEN ↵
    ↵-V=ASC(C$)
1230 RETURN
1240 REM *** GET BYTE & CHK FOR DISK ↵
    ↵-ERROR
1250 GET#5,C$:SS=ST
1260 INPUT#15,EN,EM$,ET,ES:IF EN=0 THEN ↵
    ↵-RETURN
1270 PRINT"↵DISK ERROR!↵
1280 PRINT EN;EM$;ET;ES:CLOSE 4:CLOSE 5:
    ↵-CLOSE 15
1290 REM *** MISC ROUTINES ***
1300 GOSUB 1340:GOTO 110
1310 PRINT"↵DEPRESS ANY KEY TO CONTINUE
1320 GET C$:IF C$="↵" THEN 1320
1330 RETURN
1340 PRINT"↵@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    ↵-@@@@@@@@@@@@@@@@":RETURN
1350 PRINT" (Y/N) ?  ";
1360 GOSUB 1320:IF C$<>"Y" AND C$<>"N" ↵
    ↵-THEN 1360
1370 PRINT C$ : RETURN

```

©

MICRO COMPUTER INDUSTRIES, LTD.

INVENTORY CONTROL WITH POINT OF SALE FOR CBM AND PET 32K

DISK VERSION INCLUDES: Storage of 2500 items per diskette, Accounts Receivable, Writes Purchase Orders, Invoices, Summaries, Post Income and will Sort by 10 fields.

\$100.00

INVENTORY CONTROL 8K

CASSETTE VERSION INCLUDES: Purchase Order program and Printing functions.

\$ 39.00

GENERAL LEDGER

DISK VERSION INCLUDES: The total functions of the Inventory Control programs plus; Accounts Payable, Notes Payable, Purchases, Expenses, it also issues complete Reports, Statements and Summaries.

\$350.00

Instruction Manual \$10.00 refundable with purchase.

MICRO COMPUTER INDUSTRIES, LTD.

1520 East Mulberry Suite 240
Fort Collins, Colorado 80524
1-303-221-1955